(19) Eur päisch s Patentamt
European Patent Offi e
Office européen des brevets

(11) Publication number: **0 380 099 A2**

(12) EUROPEAN PATENT APPLICATION

(21) Application number: 90101505.7

(51) Int. Cl.^{5}: G06F 7/48

(22) Date of filing: 25.01.90

(30) Priority: 27.01.89 US 303788

(43) Date of publication of application:
01.08.90 Bulletin 90/31

(84) Designated Contracting States:
DE ES FR GB

(71) Applicant: Hughes Aircraft Company
7200 Hughes Terrace P.O.Box 45066
Los Angeles California 90045-0066(US)

(72) Inventor: Davies, Steven P.
2550 S. Holmes Place
Ontario, California 91871(US)
Inventor: Espelien, Mark
5301 Vista Bonita NE
Albuquerque, New Mexico 87111(US)

(74) Representative: Witte, Alexander, Dr.-Ing.
Augustenstrasse 7
D-7000 Stuttgart 1(DE)

(54) Register and arithmetic logic unit.

(57) Register and arithmetic logic apparatus for use in processing digital data and which provides a split pipeline architecture that operates on multiple data formats. A register file (444) is employed to store data words. An arithmetic logic unit (450) processes the data words by means of two parallel arithmetic logic units (450a, 450b) that provide fixed point and floating point arithmetic processing operations, respectively. The two parallel arithmetic logic units (450a, 450b) permit processing of a plurality of predetermined data processing formats, including dual 16 bit fixed point, 32 bit fixed point, 32 bit floating point and logical data processing formats. Post-processing registers (454, 456) provide for a limiter/shifter register, a length selectable first in, first out buffer for controlling the length of the register pipeline, and logic which provides for queuing of the processed data words. The register file (444) and the fixed point arithmetic logic unit (450a) may be selectively coupled together to function as an accumulator. This function permits processing of the data words such that two 32 bit data words are accumulated into a 64 bit data word, or the dual 16 bit data words are accumulated into two 32 bit data words. Processing using the dual 16 bit format employs a pontential overflow scheme that permits a variety of signal processing algorithms to function with relatively compact code.

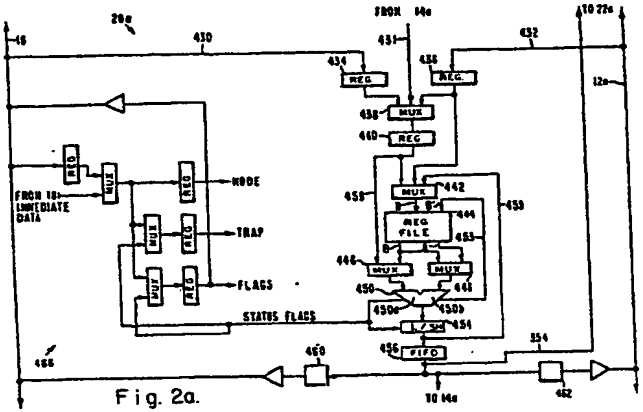


Fig. 2a.

REGISTER AND ARITHMETIC LOGIC UNIT

BACKGROUND OF THE INVENTION

The present invention generally relates to digital signal processing systems and in particular to a register and arithmetic logic unit for use in such systems that provides for a split pipeline architecture that operates on multiple data formats.

The ability to perform sophisticated vector and scalar arithmetic operations in real time is a key requirement of signal processing systems. Often, however, this requirement is also accompanied by severe physical constraints upon the size, weight, power and cooling of the signal processing system. In the past, signal processor designers have had to compromise among competing requirements, many times resulting in processors with less than adequate performance.

Conventional signal processors may also be limited in performance due to relatively slow system clock rates of around five megahertz, and limited capability to operate on 16 bit fixed point data. The fixed point operational limitations of the such conventional signal processors has become significant in many application environments. Many signal processing algorithms require arithmetic computations having a large dynamic range, making 32 bit floating point processing necessary.

Thus, in order to support the 32 bit processing requirements of state of the art signal processors, there is a need in the art for an arithmetic and register logic unit which is capable of providing fixed and floating point computations on 16 and 32 bit data.

SUMMARY OF THE INVENTION

In order to overcome some of the limitations of conventional signal processing systems, the present invention provides for a register and arithmetic logic unit which incorporates a split pipeline architecture that is capable of simultaneously operating on multiple data formats. The register and arithmetic unit comprises a register file having a data input and a plurality of data outputs which is capable of storing a plurality of data words. An arithmetic logic unit is provided which has a plurality of inputs coupled to the data outputs of the register file and which processes data words by means of two parallel arithmetic logic units that provide fixed point and floating point arithmetic processing operations, respectively. Interface logic having a plurality of data inputs for receiving data words, and a data output coupled to the register file and to the arithmetic logic means, are provided for selectively transferring data words to the register file for storage therein or to the arithmetic logic means for processing thereby.

Each input of the plurality of data inputs of the arithmetic logic unit is coupled to respective ones of the two parallel arithmetic logic units to permit processing of the input data. The data formats include dual 16 bit fixed point, 32 bit fixed point, 32 bit floating point and logical data processing formats. Post-processing registers include a limiter/shifter register, a length-select able four word first in, first out buffer that controls the length of the register pipeline, and logic which provides for queuing of the processed data words.

The register file and the fixed point arithmetic logic unit may be selectively coupled together to function as an accumulator. This function permits processing of the data words such that two 32 bit data words are accumulated into a 64 bit data word comprising two register file words, or the dual 16 bit data words are accumulated into a 32 bit data word. Processing using the dual 16 bit format employs a potential overflow scheme which permits a variety of signal processing algorithms to function with relatively compact code.

BRIEF DESCRIPTION OF THE DRAWING

The various features and advantages of the present invention may be more readily understood with reference to the following detailed description taken in conjunction with the accompanying drawing, wherein like reference numerals designate like structural elements, and in which:

FIG. 1 is a block diagram of a signal processor incorporating the register and arithmetic logic unit of the present invention;

FIG. 2a is a detailed block diagram of the register and arithmetic logic unit of the present invention;

FIG. 2b is a detailed diagram of the fixed and floating point processor of the register and arithmetic logic unit of FIG. 2a;

FIG. 3 is a detailed diagram of the register file and fixed and floating point processor of the present invention; and

FIG. 4 is a detailed diagram of the register file of the present invention.

DETAILED DESCRIPTION

Referring to FIG. 1 shown therein is a block diagram of a signal processor 10 incorporating a register and arithmetic logic unit 20 in accordance with the principles of the present invention. The signal processor 10 will be described in general terms to provide a context for the describing the register and arithmetic logic unit 20.

The signal processor 10, shown in FIG. 1, generally comprises four main sections: an input/output section, designated as I/O, a central procession unit designated as CPU, and two arithmetic elements, designated as AE0 and AE1. The input/output section includes an external interface unit 11 which provides a plurality of configurable input/output ports. The external interface unit 11 is coupled by way of data busses 12a, 12b to two data store memories 13a, 13b, that are employed to store data, and to two multipliers 14a, 14b, and two register and arithmetic logic units 20a, 20b, which operate on the data. The data store memories 13a, 13b typically store data in a predefined packed format in order to conserve memory space, in a manner which is generally known in the art.

A control store memory 15, which is employed to store control codes, is coupled by way of a control store bus 16 to an arithmetic element controller 17, to the multipliers 14a, 14b and to two register and arithmetic logic units 20a, 20b made in accordance with the principles of the present invention. A micro store memory 18 is coupled to the arithmetic element controller 17 and is employed to store microcode instructions which are utilized by the data store memories 13a, 13b, multipliers l4a, 14b, and the register and arithmetic logic units 20a, 20b.

While the present invention is disclosed with reference to its incorporation in the the above-described signal processor 10 and architecture, it is not restricted to use therewith. The present invention may be employed as a stand alone processor suitable for applications other than the above-described processor.

The processor 10 generally functions as follows. Signals to be processed by the processor 10 are received by way of the external interface unit 11 and stored in the data store memories 13a, 13b. Microcode instructions defining the processing parameters of the arithmetic elements of the processor and what steps are to be performed by the arithmetic elements, AE0, AE1, are stored in the micro store memory 18. The application program consisting of pointers to microcode instructions, programmable coefficients to be used by the arithmetic elements during computations, and intermediate data processing results from the arithmetic elements are stored in the control store memory 15. The arithmetic element controller 17 executes application programs which cause the microcode instructions to be executed and the data to be processed. The arithmetic elements AE0, AE1, operate as parallel pipeline processors, to process the data in accordance with the microcode instructions, under control of the arithmetic element controller, and in a conventionally understood manner.

Control parameters are passed from the control store memory 15 to the multipliers 14a, 14b and the register and arithmetic logic units 20a, 20b, and the data from the data store memories 13a, 13b are processed by the arithmetic elements AE0 and AE1, under control of the arithmetic element controller 17 in a conventionally understood manner.

A detailed data flow diagram of the register and arithmetic logic unit 20a of the present invention is shown in FIG. 2a. Shown therein are a plurality of data input lines to the register and arithmetic logic unit 20a comprising a control store input line 430, a multiplier input line 431 and a data store input line 432. The control store and data store input lines 430, 432 are separately coupled through control and data unpacking logic 434, 436 to a first three-input multiplexer 438. The first multiplexer 438 is coupled by way of a first register 440 to a second three-input multiplexer 442.

The second multiplexer 442 is coupled to a 32 bit by 32 word register file 444 which has two outputs A, B, that are coupled to two two-input multiplexers 446, 448, as shown. The first register 440 is also coupled to the first two input-multiplexer 446 by way of bypass path 458. The two input multiplexers 446, 448 are coupled to an arithmetic logic unit 450 which incorporates two parallel arithm tic logic units 450a, 450b that provide fixed point and floating point arithmetic processing operations, respectively.

The arithmetic logic unit 450 is coupled by way of a third two-input multiplexer 452 to limiter/shifter logic 454 and then to a four word first in, first out (FIFO) buffer 456. A first feedback loop 453 is provided between the arithmetic logic unit 450 and the register file 444, which loop is shown in more detail in FIG. 3. A second feedback loop 459 is provided from a point between the limiter/shifter logic 454 and the FIFO buffer 456 to the second thre -input multiplexer to control the signal flow through the arithmetic logic unit 20a. The first in, first out buffer is coupled to respective control store and data store memories 15, 13 by way of respective buffered packing logic units 460, 462, and to the multiplier 14a and the data store interface logic (DSIL) 22 shown in FIG. 1.

Control and timing logic 466 is provided which interfaces between the arithmetic logic unit 450 and the arithmetic element controller 17. This control and timing logic 466 provides status flags, conditional flags and other information to the controller as is identified in FIG. 2a. The control and timing logic 466 and its interconnection are clearly shown in FIG. 2a and will not be discussed in detail herein.

FIG. 2b shows a more detailed diagram of the fixed and floating point processor 450 shown in FIG. 2a. FIG. 2b shows that the processor 450 is comprised of two individual processors, namely the fixed point processor 450a and the floating point processor 450b. The two individual processors 450a, 450b are shown having their outputs coupled to the third two-input multiplexer 452.

With reference to FIG. 3 it shows a more detailed diagram of the register file 444 and fixed and floating point processor 450 of the present invention. The fixed and floating point processor 450 comprises two 16 bit fixed point incrementers 470, 472 which receive output B′ from the register file 444. Outputs of the two 16 bit fixed point incrementers 470, 472 are coupled together and provide a feedback path to the register file 444.

The output of the first two-input multiplexer 446 is coupled to one input of a 32 bit floating point arithmetic unit 474 and respective first inputs of two 16 bit fixed point arithmetic logic units 476, 478. The output of the second two-input multiplexer 446 is coupled to the second input of the 32 bit floating point arithmetic unit 474 and respective second inputs of the 16 bit fixed point arithmetic logic units 476, 478. An AND gate 480 and third and fourth two-input multiplexers 482, 484 interconnect the 16 bit fixed point arithmetic logic units 476, 478 and the two 16 it fixed point incrementers 470, 472 as shown.

Outputs of the two 16 bit fixed point arithmetic logic units 476, 478 are coupled to one input of the two-input multiplexer 452, while the output of the 32 bit floating point arithmetic unit 474 is coupled to the second input of the two-input multiplexer 452.

FIG. 4 shows a detailed diagram of the register file 444. The register file 444 includes an input multiplexer 490 which receives data inputs B and B′, and two 16 word by 32 bit registers 492, 494. Outputs A and B of the two registers 492, 494 are coupled to each of the output multiplexers 496, 498 which combine the respective A and B data to form complete A and B words, while the B output of the first register file 444 provides the B′ output.

No control lines or logic have been shown in the drawing for either the signal processor 10, or for the register and arithmetic logic unit 20 of the present invention. However,Table 1 below shows a 64 bit word having opcode mnemonics identified therein wherein the first 28 bits, identified as bits 0-27, are employed by the register and arithmetic logic unit 20 of FIG. 2. The bits identified in Table 1 as RALU are associated with the register and arithmetic logic unit 20. These opcodes are referred to in Tables 2 and 3 below, which provide a description of the microcode instruction set utilized to implement the control logic for the register and arithmetic logic unit 20 of the present invention.

Table 1.

Word Partitioning			
Unit	Bits	Field	C mm nt
Address Gen.	63-57	AOP	
	56-53	OP1	Data/Offset
	52-49	OP2	Data/Offset
	48-46	CR	
	45-43	DR	
Memory	42-39	CS	
	38-36	DS	
Multiplier	35-34	A	Status/Mode Register (SMR)
	33-31	B	SMR
	30-28	MOP	SMR
RALU	27-26	I	SMR
	25	FLG	SMR
	24-21	MD	SMR
	20-15	RPO	SMR
	14-10	A	SMR
	9-5	B	SMR
	4	S	SMR
	3-2	DE	
	1,0	FI,FO	Reserved

Table 2. Microcode Instruction Set

Field		Opcode	Description
I	0	H	No operation
	1	M	Load RIR from multiplier output
	2	C	Load RIR from CSBUS
	3	D	Load RIR from DSBUS
F	0	H	No operation
	1	L	Load the SMR flags with the RALU flags
MD		UN	Unconditional operation
		EX	Extended precision operation
		LT	Do operation if LT flag = 1
		GE	Do operation if LT flag = 0
		EQ	Do operation if EQ flag = 1
		NE	Do operation if EQ flag = 0
		GT	Do operation if GT flag = 1
		LE	Do operation if GT flag = 0
		CO	Do operation if CO flag = 1
		NC	Do operation if CO flag = 0
		FL	Convert FX32 to FL32
		FX	Convert FL32 to FX32

		PO	IF TRPO is true
			do conditional operation
			set SMR bit 2 (RLIM)
			ELSE
			do default operation
			clear SMR bit 2 (RLIM)
			Clear TRPO

		L1	Toggle bit 1 in SMR (RLIM)
		L2	Toggle bit 0 in SMR (RLIM)
		CR	Toggle RCSR mode
		DR	Toggle RDSR mode
		CD	Toggle RCSD mode
S	0	H	No limiter/shifter operation on result
S	1	S	Perform limiter/shifter operation on RALU result as defined in SMR

Table 2. Microcode Instruction Set (Con't)

Field	Opcode	Description
ROP	MOV	Data Move - Logical
	REPM	Replicate MSW - Logical
	REPL	Replicate LSW - Logical
	SWP	Swap MSW and LSW - Logical
	AND	Boolean AND - Logical
	XOR	Boolean Exclusive OR - Logical
	ORB	Boolean OR - Logical
	ZERO	Pass zero - Logical
	XMOV	Data move - FX32
	XNEG	Negate - FX32
	XINC	Increment by 1 - FX32
	XDEC	Decrement by 1 - FX32
	XADD	Addition - FX32
	XSUB	Subtraction - FX32
	XSBR	Reverse subtraction - FX32
	FMOV	Data move - FL32
	FNEG	Negate - FL32
	FDBL	Multiply by 2 - FL32
	FHLF	Divide by 2 - FL32
	FADD	Addition - FL32
	FSUB	Subtraction - FL32
	FSBR	Reverse subtraction - FL32
	CONV	Format conversion
	QMOV	Data move - Q32
	QNE	Negate - Q32
	QINC	Increment by 1 - Q32
	QDEC	Decrement by 1 - Q32
	QAA	Addition - Q32
	QSS	Subtraction - Q32
	QAS	Addition/Subtraction - Q32
	QSA	Subtraction/Addition - Q32

Table 2. Microcode Instruction Set (Con't)

Field		Opcode	Description
DE	0	H	No operation
	1	Bj	Load RALU result to Bj
	2	Ij	Load RALU direct to B
	3	Dj	Load DS bus direct to Bj
FI	0	H	No operation
	1	L	Load the RALU result into the FIFO Increment to input pointer modulo 4 If FIFO is full, increment output pointer modulo 4
FO	0	H	No operation
	1	U	If FIFO is not empty, increment output pointer modulo 4

Table 3.

Status-Mode Register		
SMR Bit	Name	Description
23	TRPO	Trap: RALU potential overflow
22	TRIR	Trap: RALU inexact result
21	TROF	Trap: RALU overflow
20	TRIO	Trap: RALU invalid operation
19	TRUF	Trap: RALU underflow
18	.....	Spare bit
17	RLTM	Flag: RALU Less than zero, MSW
16	REQM	Flag: RALU Equal to zero, MSW
15	RGTM	Flag: RALU Greater than zero, MSW
14	RCOM	Flag: RALU Carry/Overflow/Nan, MSW
13	RLTL	Flag: RALU Less than zero, LSW
12	REQL	Flag: RALU Equal to zero, LSW
11	RGTL	Flag: RALU Greater than zero, LSW
10	RCOL	Flag: RALU Carry/Overflow.Nan, LSW
9-8	RRND	Mode: RALU rounding
7	.....	Spare bit
6	RRND	Mode RALU CS delay
5	RCSR	Mode RALU CS short-word read
4	RDSR	Mode RALU DS short-word read
3	RIDX	Mode RALU external index enable
2-0	RLIM	Mode RALU limiter/shifter

From the information provided in Tables 1, 2 and 3, one skilled in the art of signal processor design can manually, or by m ans of commercially available software, such as software available from The CAD Group of Soquel, California, produce the timing and control logic which implements the register and arithmetic logic unit 20 of the present invention. This control logic, along with the data flow description provided herein, and its description of operation, is sufficient to permit one skilled in the art to construct the present invention.

The register and arithmetic logic unit 20a operates as follows. An applications program is loaded into the control store memory 15 and data to be processed is stored in the data store memory 13a. Microcode instructions are stored in the micro store memory 18 while control parameters are stored in the control store memory 15. Under control of the applications program, and in accordance with the microcode instructions from the micro stor memory 18, control and data signals are coupled to respective portions of each of the arithmetic elements, AE0, AE1, for processing.

With reference to the arithmetic and register logic unit 20a, it receives signals along control store and data store busses 16, 12a and from the multiplier 14a, and sequentially loads the data into the register file 444 for storage. The fixed point and floating point logic units 450a, 450b of the arithmetic logic unit 450 operate in parallel as split pipeline processor to operate on the data stored in the register file 444.

The arithmetic logic unit 20a operates on data having four distinct types of data formats. These formats include dual 16 bit fixed point, 32 bit fixed point, 32 bit floating point and logical data processing formats. By appropriate coding of the applications program, selected ones of the four data formats may be processed at one time the arithmetic elements, and each arithmetic element can perform similar processing on different data at the same time. In addition, by bypassing the register file 444, and passing data directly to the arithmetic logic unit 450 allows two operands to be brought into the register and arithmetic logic unit 10 at one time, one through the register file 444, and the other over the bypass path 453.

FIG. 3 shows the details of the structure which provides for operation on data having the four different data formats. The multiplexers 446, 448 select a 32 bit carry path or 64 bit carry path for dual 16 bit fixed point or 32 bit fixed point operations, respectively. The gate 480 prevents carry into the first 16 bit fixed point arithmetic logic unit 498 which processes the most significant word, when operating in the dual 16 bit fixed point mode.

The register file 444 and the fixed point arithmetic logic unit 450a may also be selectively linked to operate as an accumulator. This permits processing of data words such that two 32 bit data words are accumulated into a 64 bit data word comprising two register file words, or dual 16 bit data words are accumulated into two 32 bit data words.

The dual 16 bit format employs a potential overflow scheme which permits a variety of signal processing algorithms to function with relatively compact code. In this scheme the two halves of the dual 16 bit format are treated as the real and imaginary portions of a complex value. With each portion representing a fractional value, a potential overflow is signaled if the complex magnitude of the value would be greater than 0.5.

At the end of a vector operation, that is, an operation on an array of complex values, the potential overflow flag may be examined and used to set the mode of the limiter/shifter. If a potential overflow has been signaled then the limiter/shifter mode should be set to divide all the vector elements by two(an arithmetic right shift) on the next vector operation, thereby preventing any overflow conditions. If a block exponent is incremented each time the potential overflow flag is checked, the potential overflow flag can then be used to implement a very efficient from of block floating point.

Thus there has been described a new and improved a register and arithmetic logic unit that provides for a split pipeline architecture that operates on multiple data formats. These formats include dual 16 bit fixed point, 32 bit fixed point, 32 bit floating point and logical data processing formats. Post-processing registers including a limiter/shifter register and a length-selectable FIFO buffer which control the length of the pipeline, and logic which provides for queuing of the processed data words.

It is to be understood that the above-described embodiment is merely illustrative of some of the many specific embodiments which represent applications of the principles of the present invention. Clearly, numerous and other arrangements can be readily devised by those skilled in the art without departing from the scope of the invention.

Claims

1. A register and arithmetic logic unit comprising a register file (444) having a plurality of data inputs and a plurality of data outputs, and adapted to store a plurality of data words, characterized by:

- arithmetic logic means (450) having a plurality of inputs coupled to the data outputs of the register file (444) and having a data output, for processing data words by means of two parallel arithmetic logic units (450a, 450b) which provide fixed point and floating point arithmetic processing operations, respectively; and
- interface logic means having a plurality of data inputs (430, 431, 432) for receiving data words to be processed coupled to the data inputs of the register file (444) and arithmetic logic means (450), for selectively transferring data words to the register file (444) for storage therein or transferring data words to the arithmetic logic means (450) for processing thereby.

2. The register and arithmetic logic unit of claim 1, charact rized by:
- post-processing m ans comprising a limiter/shifter logic circuit (454) coupled to th data output of the arithmetic logic means (450) for providing limiting and scaling of the data words processed by the two parallel arithmetic logic units (450a, 450b).

3. The register and arithmetic logic unit of claim 2, wherein the post-processing means is characterized by:
- a first-in, first-out buffer (456) serially coupled to the limiter/shifter logic circuit (454) for controlling the length of the pipeline of the register and arithmetic logic unit (20a) and for providing queuing of the processed data words.

4. The register and arithmetic logic unit of any of claims 1 - 3, characterized by:
- a limiter/shifter logic circuit (454) coupled to the data output of the arithmetic logic means (450) and having an output coupled to an input of the register file means (444) ; and
- a first-in, first-out buffer (456) having an input coupled to the output of the limiter/shifter logic circuit (454) and having an output comprising the output of the register and arithmetic logic unit (20a);
- wherein the limiter/shifter logic circuit (454) and the first-in, first-out buffer (456) provide for limiting and scaling of the data words processed by the arithmetic logic means (450) and queuing of the processed data words.

5. The register and arithmetic logic unit of any of claims 1 - 4, characterized by each input of the plurality of data inputs of the arithmetic logic means (20a) being coupled to respective ones of the two parallel arithmetic logic units (450a, 450b) to permit processing of a plurality of predetermined data processing formats.

6. The register and arithmetic logic units of any of claims 1 - 5, characterized by the register file (444) and the fixed point arithmetic logic unit (450a) being configured as an accumulator.

7. The register and arithmetic logic unit of any of claims 1 - 6, characterized by the two parallel arithmetic logic units (450a, 450b) providing for processsing of the data in dual 16 bit fixed point, 32 bit fixed point, 32 bit floating point and logical processing formats.

8. The register and arithmetic logic unit of any of claims 1 - 7, characterized by the register file (444) and the fixed point arithmetic logic (450a) unit being configured as an accumulator to provide for processing of the data words such that a 32 bit data word is accumulated into a 64 bit data word comprising two register file words or the dual 16 bit words are accumulated into two 32 bit data words.

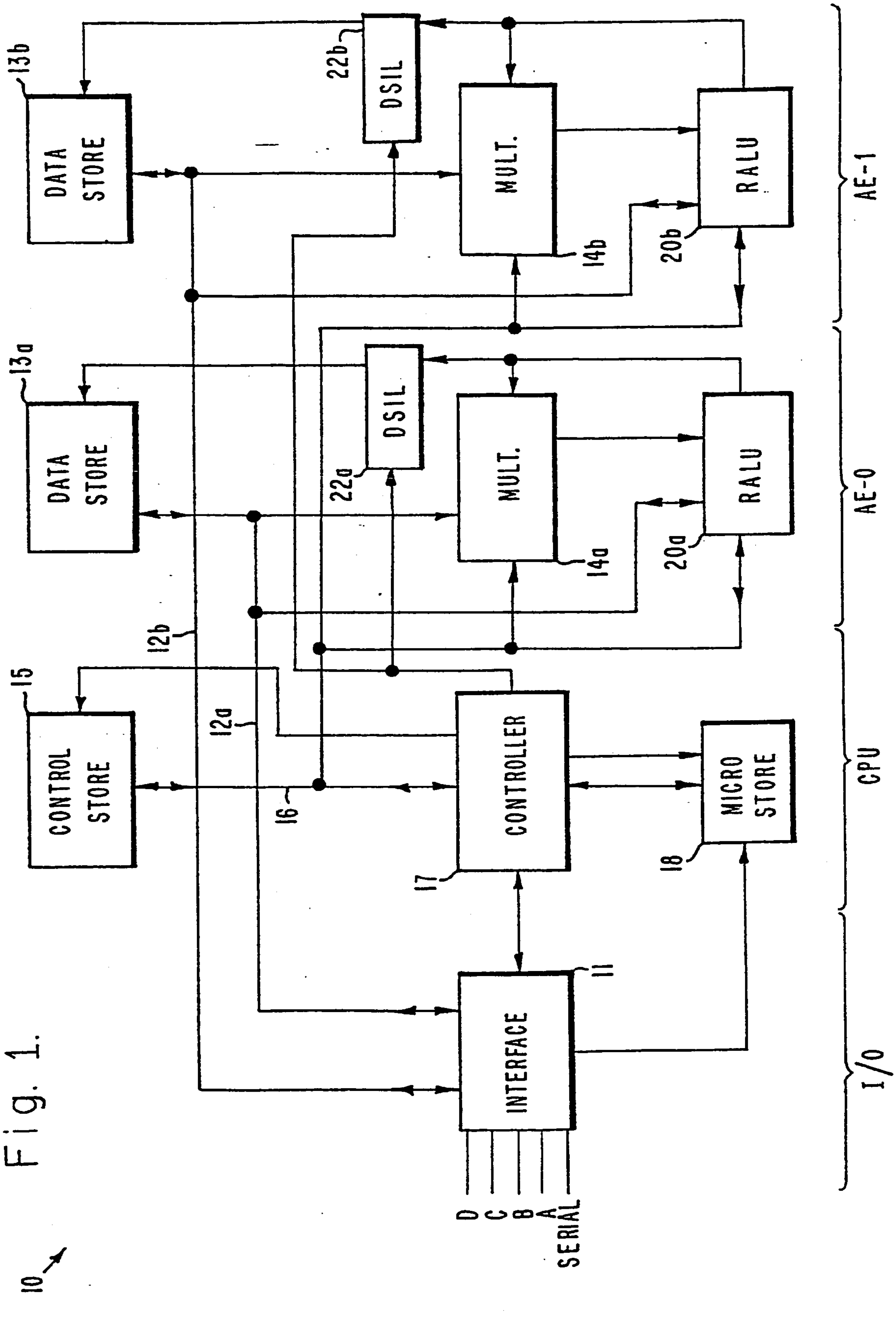

DATA STORE
13b
DSIL
22b
MULT.
14b
RALU
20b
AE-1
DATA STORE
13a
DSIL
22a
MULT.
14a
RALU
20a
AE-0
CONTROL STORE
15
12b
12a
16
CONTROLLER
17
MICRO STORE
18
CPU
INTERFACE
11
I/O
D
C
B
A
SERIAL
10

Fig. 1.

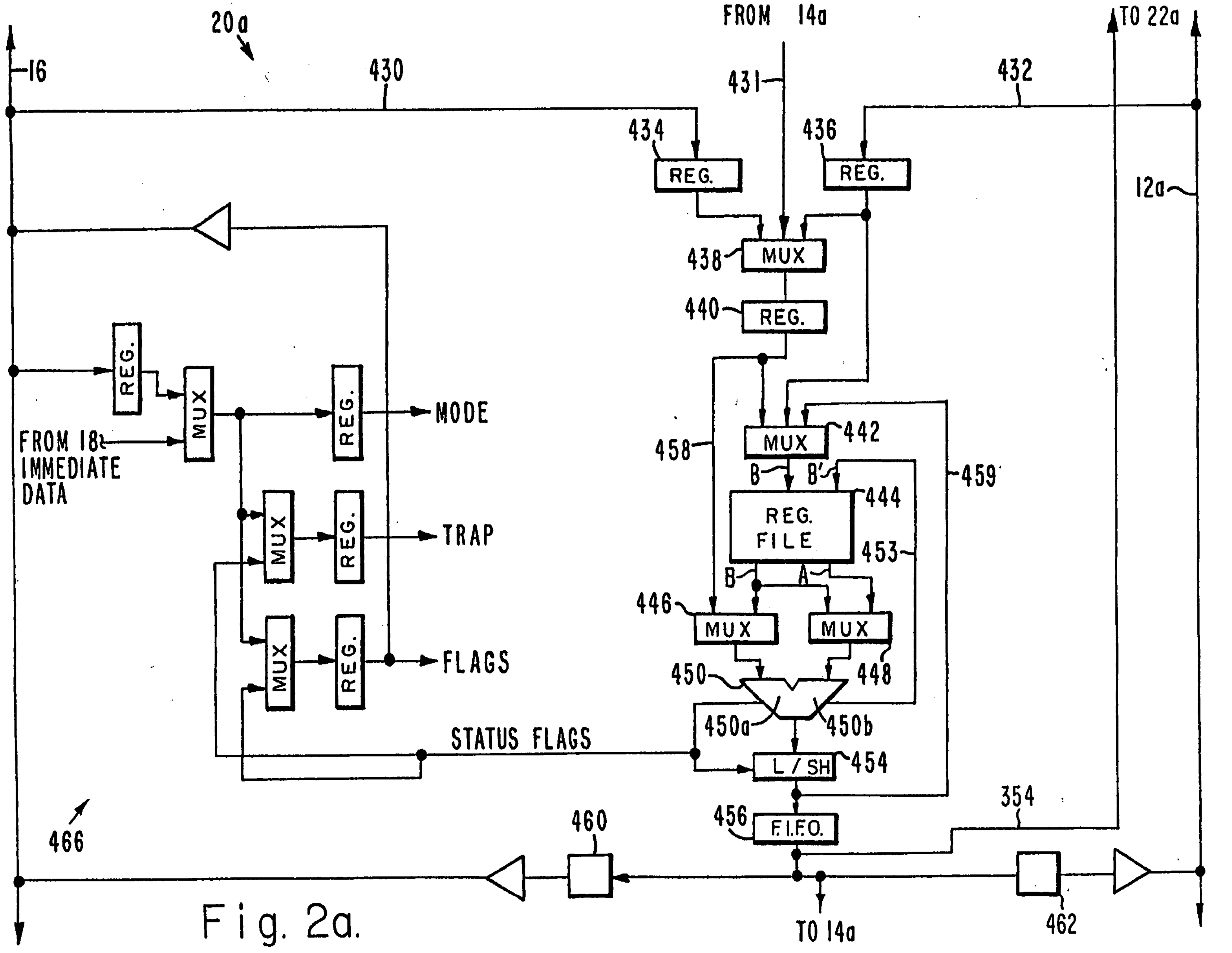

20a
16
430
FROM 14a
431
432
TO 22a
12a
434
REG.
436
REG.
438
MUX
440
REG.
458
MUX
442
B
B'
459
444
REG. FILE
453
B
A
446
MUX
MUX
448
450
450a
450b
STATUS FLAGS
L / SH
454
354
456
F.I.F.O.
460
462
TO 14a
466
REG.
MUX
FROM 18
IMMEDIATE DATA
REG.
MODE
MUX
REG.
TRAP
MUX
REG.
FLAGS

Fig. 2a.

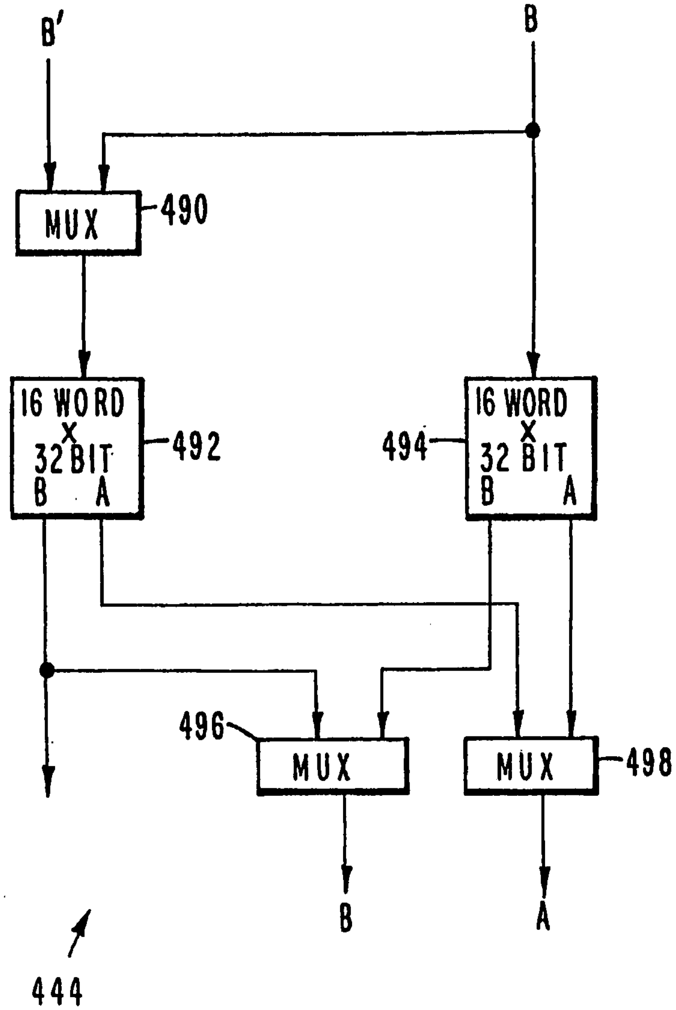

B'
B
MUX
490
16 WORD
X
32 BIT
B A
492
494
16 WORD
X
32 BIT
B A
496
MUX
MUX
498
B
A
444

Fig. 4.

Fig. 2b.

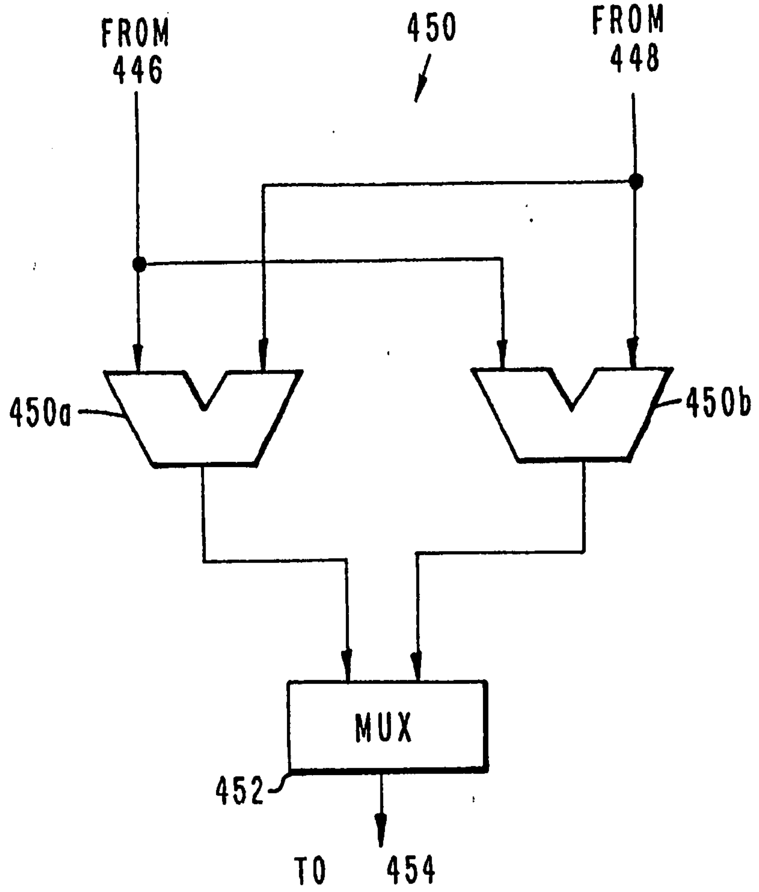
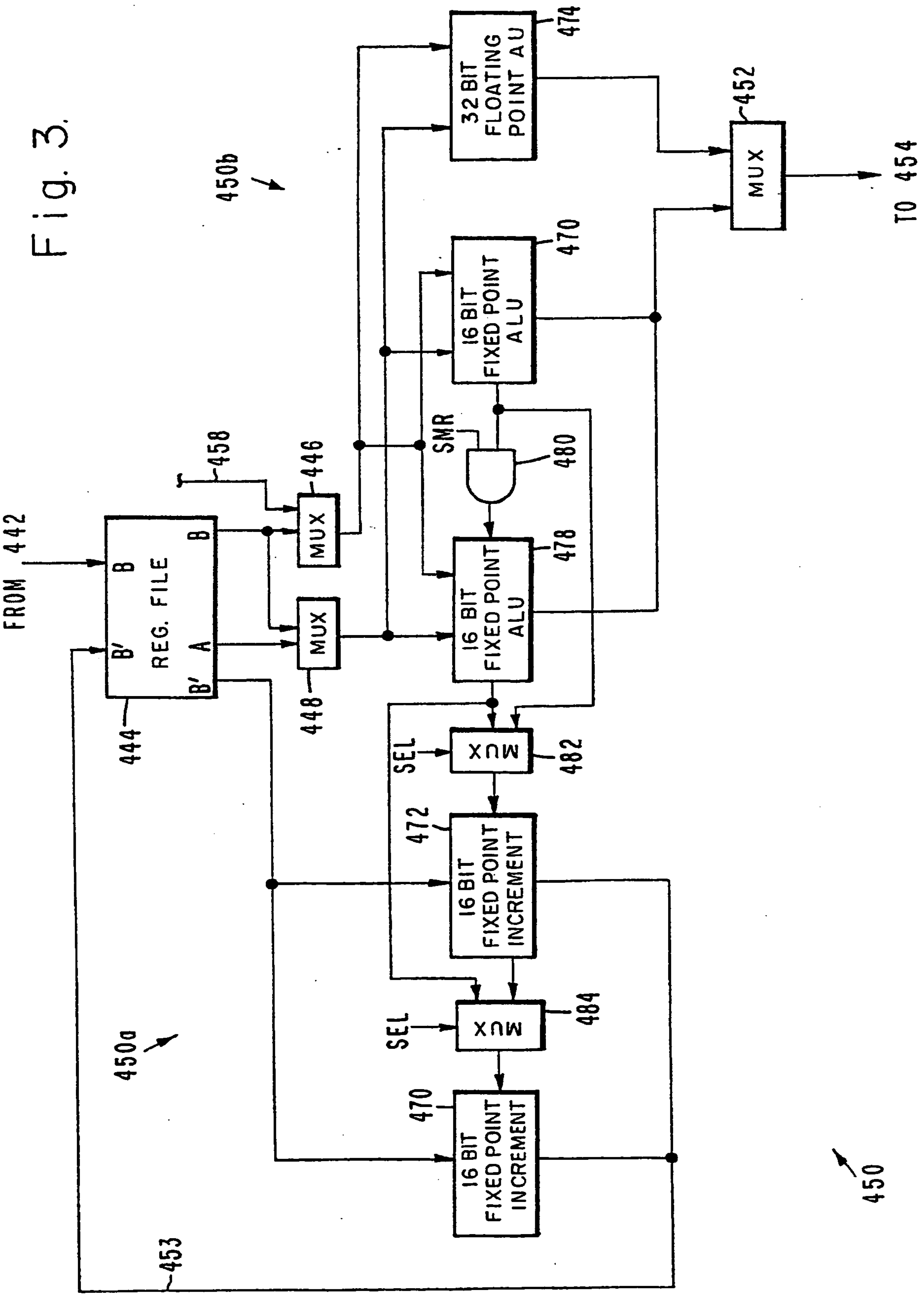

FROM
446
450
FROM
448
450a
450b
MUX
452
TO 454

Fig. 3


450b
32 BIT FLOATING POINT AU
474
452
MUX
TO 454
16 BIT FIXED POINT ALU
470
SMR
480
458
446
MUX
FROM 442
B
REG. FILE
B'
A
B'
B
444
MUX
448
16 BIT FIXED POINT ALU
478
SEL
MUX
482
472
16 BIT FIXED POINT INCREMENT
SEL
MUX
484
450a
470
16 BIT FIXED POINT INCREMENT
450
453

(19) Eur päisches Patentamt
European Patent Office
Offic uropéen des brevets

(11) Publication number: 0 380 099 A3

(12) EUROPEAN PATENT APPLICATION

(21) Application number: 90101505.7

(51) Int. Cl.^{5}: G06F 7/48

(22) Date of filing: 25.01.90

(30) Priority: 27.01.89 US 303788

(43) Date of publication of application:
01.08.90 Bulletin 90/31

(84) Designated Contracting States:
DE ES FR GB

(88) Date of deferred publication of the search report:
24.06.92 Bulletin 92/26

(71) Applicant: Hughes Aircraft Company
7200 Hughes Terrace P.O. Box 45066
Los Angeles, California 90045-0066(US)

(72) Inventor: Davies, Steven P.
2550 S. Holmes Place
Ontario, California 91871(US)
Inventor: Espelien, Mark
5301 Vista Bonita NE
Albuquerque, New Mexico 87111(US)

(74) Representative: Witte, Alexander, Dr.-Ing.
Augustenstrasse 7
W-7000 Stuttgart 1(DE)

(54) Register and arithmetic logic unit.

(57) Register and arithmetic logic apparatus for use in processing digital data and which provides a split pipeline architecture that operates on multiple data formats. A register file (444) is employed to store data words. An arithmetic logic unit (450) processes the data words by means of two parallel arithmetic logic units (450a, 450b) that provide fixed point and floating point arithmetic processing operations, respectively. The two parallel arithmetic logic units (450a, 450b) permit processing of a plurality of predetermined data processing formats, including dual 16 bit fixed point, 32 bit fixed point, 32 bit floating point and logical data processing formats. Post-processing registers (454, 456) provide for a limiter/shifter register, a length selectable first in, first out buffer for controlling the length of the register pipeline, and logic which provides for queuing of the processed data words. The register file (444) and the fixed point arithmetic logic unit (450a) may be selectively coupled together to function as an accumulator. This function permits processing of the data words such that two 32 bit data words are accumulated into a 64 bit data word, or the dual 16 bit data words are accumulated into two 32 bit data words. Processing using the dual 16 bit format employs a pontential overflow scheme that permits a variety of signal processing algorithms to function with relatively compact code.

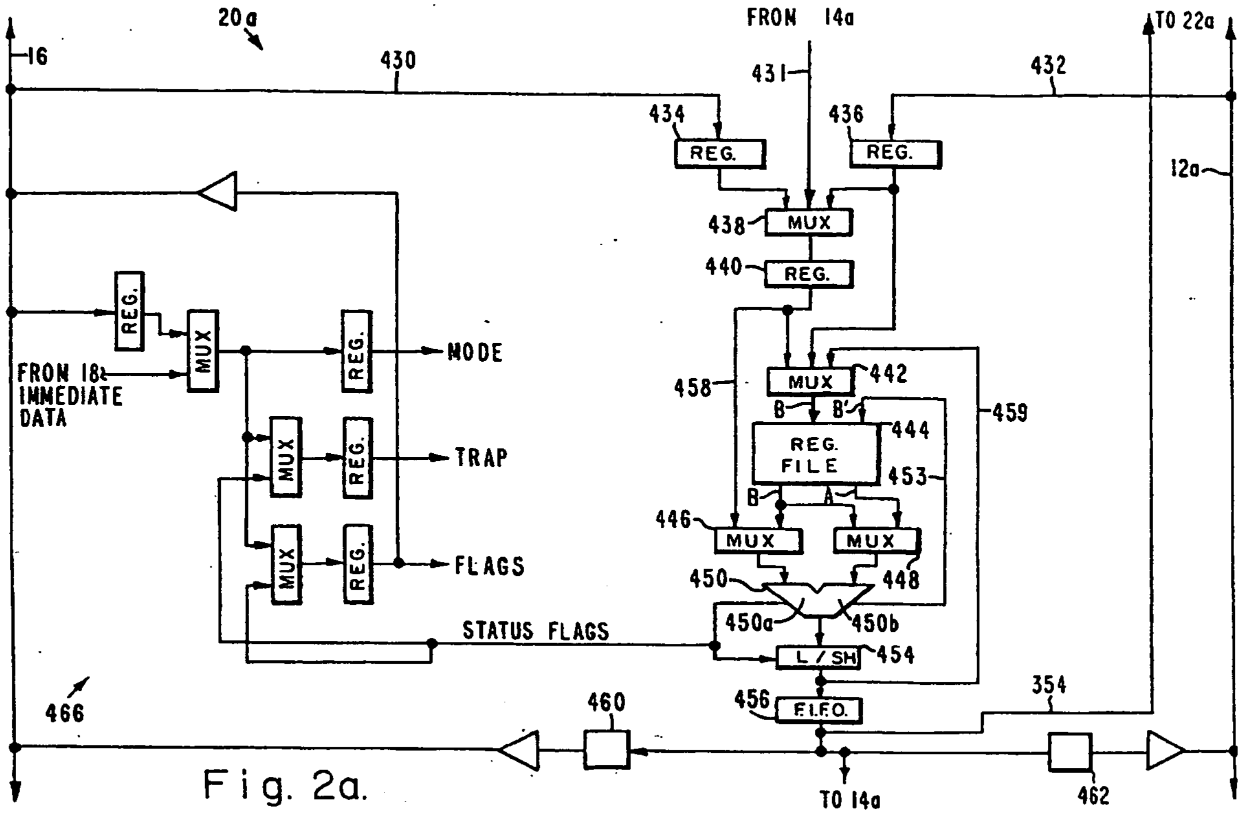

20a
16
430
FROM 14a
431
432
TO 22a
434
REG.
436
REG.
12a
438
MUX
440
REG.
REG.
MUX
REG.
MODE
FROM 18a
IMMEDIATE
DATA
458
MUX
442
459
B
B'
444
REG.
FILE
MUX
REG.
TRAP
453
B
A
446
MUX
MUX
448
MUX
REG.
FLAGS
450
450a
450b
STATUS FLAGS
L / SH
454
466
460
456
F.I.F.O.
354
TO 14a
462

Fig. 2a.

European Patent Office

EUROPEAN SEARCH REPORT

Application Number

EP 90 10 1505
Page 1

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. Cl.5)
Y A	EP-A-0 067 667 (DATA GENERAL) * page 2, line 30 - page 3, line 12 * * page 4, line 26 - page 5, line 2; figures 2,4 * ---	1-6 8	G06F7/48
Y A	CONFERENCE RECORD OF THE GLOBAL TELECOMMUNICATIONS CONFERENCE 1987 vol. 1, November 1987, TOKYO pages 442 - 446; H. GAMBE ET AL.: 'A 32 BIT FLOATING POINT DIGITAL SIGNAL PROCESSOR FDSP-4 AND ITS APPLICATION TO THE COMMUNICATION SYSTEMS' * abstract; figure 3 * * paragraph 3.2 * ---	1-6 7	
A	ELECTRONIC DESIGN. vol. 31, no. 16, August 1983, HASBROUCK HEIGHTS, NEW JERSEY pages 133 - 138; J.ELDON ET AL.: 'Floating-point chips carve out FFT systems' * figures 2,3 * ---	1,2,4,6	
			TECHNICAL FIELDS SEARCHED (Int. Cl.5)
A	IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS. vol. 35, no. 5, May 1988, NEW YORK US pages 495 - 505; B.BARAZESH ET AL.: 'A VLSI Signal Processor with Complex Arithmetic Capability' * page 496, left column, line 13 - line 15 * * page 496, right column, line 18 - line 20; figure 9 * --- -/--	3,4,7,8	G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	09 APRIL 1992	P.H.W. Verhoof

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

European Patent
Office

EUROPEAN SEARCH REPORT

Application Number

EP 90 10 1505
Page 2

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int. Cl.5)
A	CONFERENCE RECORD OF THE TWENTY-SECOND ASILOMAR CONFERENCE ON SIGNALS, SYSTEMS & COMPUTERS vol. 1, November 1988, PACIFIC GROVE, CALIFORNIA pages 898 - 902; N.BEDARD ET AL.: 'The Weitek 64-bit Floating-Point Datapath Unit' * page 900, left column, line 1 - line 17 * * page 900, right column, line 23 - page 901, line 13; figures 1,2 * -----	1,5,6	
			TECHNICAL FIELDS SEARCHED (Int. Cl.5)

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	09 APRIL 1992	P.H.W. Verhoof

CATEG RY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
: non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document